DETECTING WI-FI ATTACKS WITH KISMET

Wireless Deauthentication Attacks: Creation & Detection Using Kismet

Name: Sarah Rachel

Introduction to Kismet:

- Kismet is a leading open-source wireless network detector & sniffer
- Supports Wi-Fi, Bluetooth, Zigbee & other RF protocols.
- Used for:
    - Network discovery
    - Security auditing
    - Penetration testing.
- Runs on Linux, macOS & has limited Windows support (via remote capture).
- Passive monitoring tool — captures traffic without transmitting packets, ensuring stealthy detection.
- Detects visible and hidden wireless networks, rogue devices & security vulnerabilities.
- Free and open-source, licensed under the GNU GPL.

Key Features & Capabilities

- Supports Wi-Fi (802.11), Bluetooth & other RF signals (including hidden SSIDs).
- Real-time packet capture and logging (PCAP, JSON formats; Wireshark-compatible).
- Alerts for wireless attacks (e.g., Deauthentication floods, rogue APs, etc)
- Automated channel hopping and GPS integration for wardriving and network mapping.
- Extensible with plugins/modules for advanced security analytics and custom protocol support.

Creating a Deauth Attack Demo

Overview of Demo Environment Setup using either of the below options for getting the packets and then analyzing

PCAP Replay Approach	Lab Environment	Live Attack
Simulates a deauth attack using pre-captured Wi-Fi traffic files (no special hardware required)	Recommended for ethical, isolated testing using virtual machines or test networks.	Requires compatible Wi-Fi adapter (Tp-link Ac1300) with monitor mode and is only legal on network with permission

Required Tools and Files

- **Kismet:** Intrusion detection and wireless monitoring tool (for attack detection).
- **PCAP File:** Pre-recorded Wi-Fi capture that includes deauthentication attack frames.
- Alternative Attack Tools (for real lab):
  - wifi-deauth (Python tool for sending deauth packets)
  - aireplay-ng (from Aircrack-ng suite, for generating deauth attacks)
- **WSL-Kali Linux:** Environment for running tools and Kismet dashboard.

Installation of Kali Linux on to WSL :

wsl --set-default-version 2

--install -d kali-linux

```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>wsl --install
A distribution with the supplied name already exists. Use --name to chose a different name.
Error code: Wsl/InstallDistro/ERROR_ALREADY_EXISTS

C:\Windows\System32>
C:\Windows\System32>wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
The operation completed successfully.

C:\Windows\System32>wsl --install -d kali-linux
Downloading: Kali Linux Rolling
Installing: Kali Linux Rolling
Distribution successfully installed. It can be launched via 'wsl.exe -d kali-linux'
Launching kali-linux...
Waiting for systemd to start...
running
Please create a default Kali WSL user. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: sarah
New password:
Retype new password:
passwd: password updated successfully
usermod: no changes
┌──(sarah㉿SARAH)-[/mnt/c/Windows/System32]
└─$
```

Installation of Kismet on to Kali Linux :

sudo apt install kismet

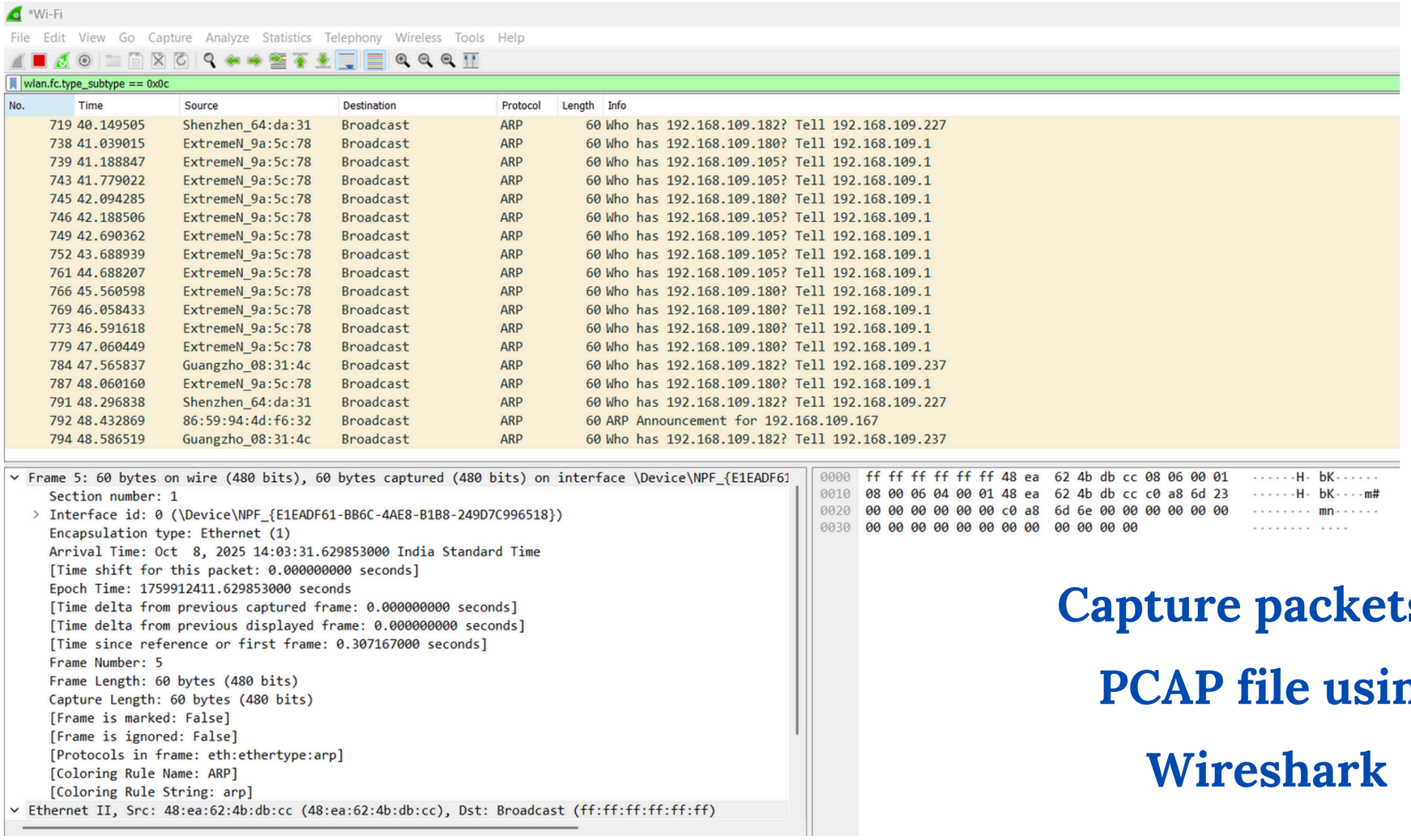
```
┌──(sarah㉿SARAH)-[/mnt/c/Windows/System32]
└─$ sudo apt install kismet
Installing:
  kismet

Installing dependencies:
  kismet-capture-common             libblas3           libwebsockets19t64
  kismet-capture-hak5-wifi-coconut  libbtbb1           python3-bluepy
  kismet-capture-linux-bluetooth    libgfortran5       python3-kismetcapturebtgeiger
  kismet-capture-linux-wifi         libglib2.0-0t64    python3-kismetcapturefreaklabszigbee
  kismet-capture-nrf-51822          libglib2.0-data    python3-kismetcapturertl433
  kismet-capture-nrf-52840          liblapack3         python3-kismetcapturertladsb
  kismet-capture-nrf-mousejack      libnl-3-200        python3-kismetcapturertlamr
  kismet-capture-nxp-kw41z          libnl-genl-3-200   python3-numpy
  kismet-capture-rz-killerbee       libnm0             python3-numpy-dev
  kismet-capture-ti-cc-2531         libprotobuf32t64   python3-protobuf
  kismet-capture-ti-cc-2540         librtlsdr0         python3-serial
  kismet-capture-ubertooth-one      libsensors-config  python3-websockets
  kismet-core                       libsensors5        shared-mime-info
  kismet-logtools                   libubertooth1      xdg-user-dirs
  libatomic1                        libusb-1.0-0

Suggested packages:
  gpsd           festival            python-bluepy-doc  python-numpy-doc  python3-wxgtk3.0
  kismet-doc     low-memory-monitor  gcc                python3-dev       | python3-wxgtk
  kismet-plugins lm-sensors          gfortran           python3-pytest

Summary:
  Upgrading: 0, Installing: 45, Removing: 0, Not Upgrading: 0
  Download size: 27.1 MB
  Space needed: 101 MB / 1,025 GB available

Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-last-snapshot/main amd64 python3-numpy-dev amd64 1:2.2.4+ds-1 [139 kB]
Get:2 http://kali.download/kali kali-last-snapshot/main amd64 libblas3 amd64 3.12.1-6 [160 kB]
Get:3 http://kali.download/kali kali-last-snapshot/main amd64 libgfortran5 amd64 15.2.0-1 [862 kB]
Get:4 http://kali.download/kali kali-last-snapshot/main amd64 liblapack3 amd64 3.12.1-6 [2,447 kB]
Get:5 http://http.kali.org/kali kali-last-snapshot/main amd64 python3-numpy amd64 1:2.2.4+ds-1 [5,096 kB]
Get:6 http://kali.download/kali kali-last-snapshot/main amd64 kismet-capture-common all 2023.07.R2-0kali2 [13.5 kB]
Get:7 http://kali.download/kali kali-last-snapshot/main amd64 libusb-1.0-0 amd64 2:1.0.29-2 [59.7 kB]
Get:8 http://kali.download/kali kali-last-snapshot/main amd64 libwebsockets19t64 amd64 4.3.5-1 [231 kB]
Get:9 http://kali.download/kali kali-last-snapshot/main amd64 kismet-capture-hak5-wifi-coconut amd64 2023.07.R2-0kali2 [89.0 kB]
Get:10 http://kali.download/kali kali-last-snapshot/main amd64 kismet-capture-linux-bluetooth amd64 2023.07.R2-0kali2 [49.8 kB]
Get:11 http://kali.download/kali kali-last-snapshot/main amd64 libatomic1 amd64 15.2.0-1 [9,492 B]
Get:12 http://kali.download/kali kali-last-snapshot/main amd64 libglib2.0-0t64 amd64 2.84.4-3 [1,518 kB]
Get:13 http://kali.download/kali kali-last-snapshot/main amd64 libnl-3-200 amd64 3.7.0-2 [59.4 kB]
Get:14 http://kali.download/kali kali-last-snapshot/main amd64 libnl-genl-3-200 amd64 3.7.0-2 [18.1 kB]
```

Step-by-Step Attack Simulation (PCAP Replay)

1. **Obtain a Deauth PCAP:** Download a sample .pcap file that contains deauthentication attack frames.
2. **Transfer PCAP to Environment:** Place the file in your Linux/WSL filesystem for easy access.
3. **Launch Kismet in Replay Mode:**
    a. **Run:** kismet -c /path/to/deauth_attack.pcap
4. **Open Kismet Dashboard:**
    a. **Access** http://localhost:2501 in browser.
5. **Analyze Alerts:**
    a. Look for alerts such as "Deauthentication flood" in the Alerts/Events tab.
    b. Review affected client and AP MAC addresses, event timeline, and severity.

Live/Active Attack Steps (if hardware available)

1. Put Wi-Fi adapter into monitor mode.
2. Use wifi-deauth or aireplay-ng to send deauth packets to the test AP.
3. Observe real-time events and alerts in Kismet as the attack is detected.

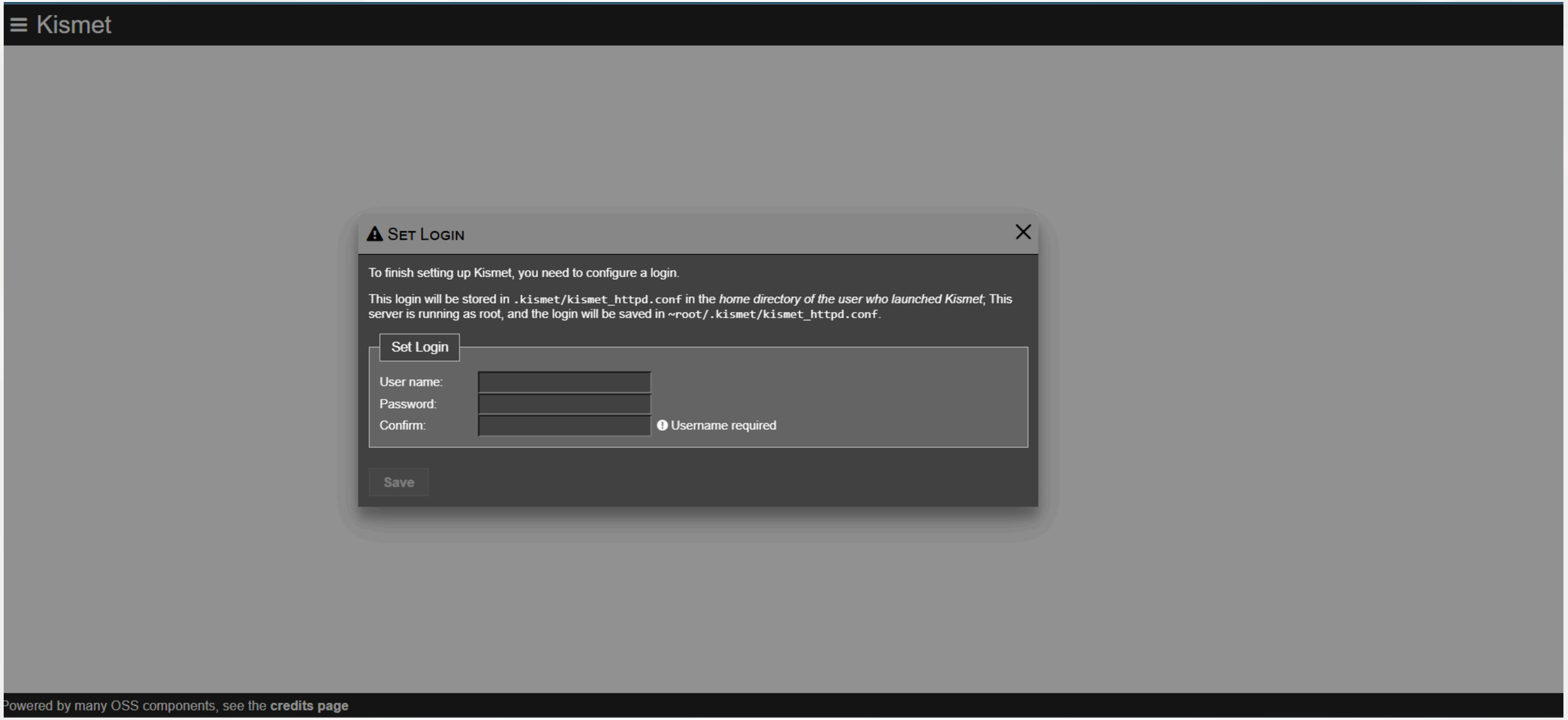


Capture packets to PCAP file using Wireshark

Replay the PCAP file:

kismet -c ~/wpa-Induction.pcap

```
┌──(wifi-deauth-venv)(sarah㉿SARAH)-[~/pcap_samples/wifi-deauth]
└─$ cp /mnt/c/Users/dell/Desktop/Sarah\ Rachel/MIT/wpa-Induction.pcap ~/

┌──(wifi-deauth-venv)(sarah㉿SARAH)-[~/pcap_samples/wifi-deauth]
└─$ kismet -c ~/wpa-Induction.pcap
INFO: Including sub-config file: /etc/kismet/kismet_httpd.conf
INFO: Including sub-config file: /etc/kismet/kismet_memory.conf
INFO: Including sub-config file: /etc/kismet/kismet_alerts.conf
INFO: Including sub-config file: /etc/kismet/kismet_80211.conf
INFO: Including sub-config file: /etc/kismet/kismet_logging.conf
INFO: Including sub-config file: /etc/kismet/kismet_filter.conf
INFO: Including sub-config file: /etc/kismet/kismet_uav.conf
INFO: Loading config override file '/etc/kismet/kismet_package.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_package.conf
INFO: Loading config override file '/etc/kismet/kismet_site.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_site.conf
INFO: Local config and cache directory '/home/sarah/.kismet/' does not
      exist; creating it.
KISMET - Point your browser to http://localhost:2501 (or the address of this system) for the Kismet UI
       'groups' command.
ERROR: Data source '/home/sarah/wpa-Induction.pcap / /home/sarah/wpa-Induct
       ion.pcap' could not launch IPC helper
ERROR: IPC cannot run binary '/usr/bin/kismet_cap_ti_cc_2531', Kismet was
       installed setgid and you are not in that group. If you recently
       added your user to the kismet group, you will need to log out and
       back in to activate it.  You can check your groups with the
       'groups' command.
```



Kismet dashboard - http://localhost:2501

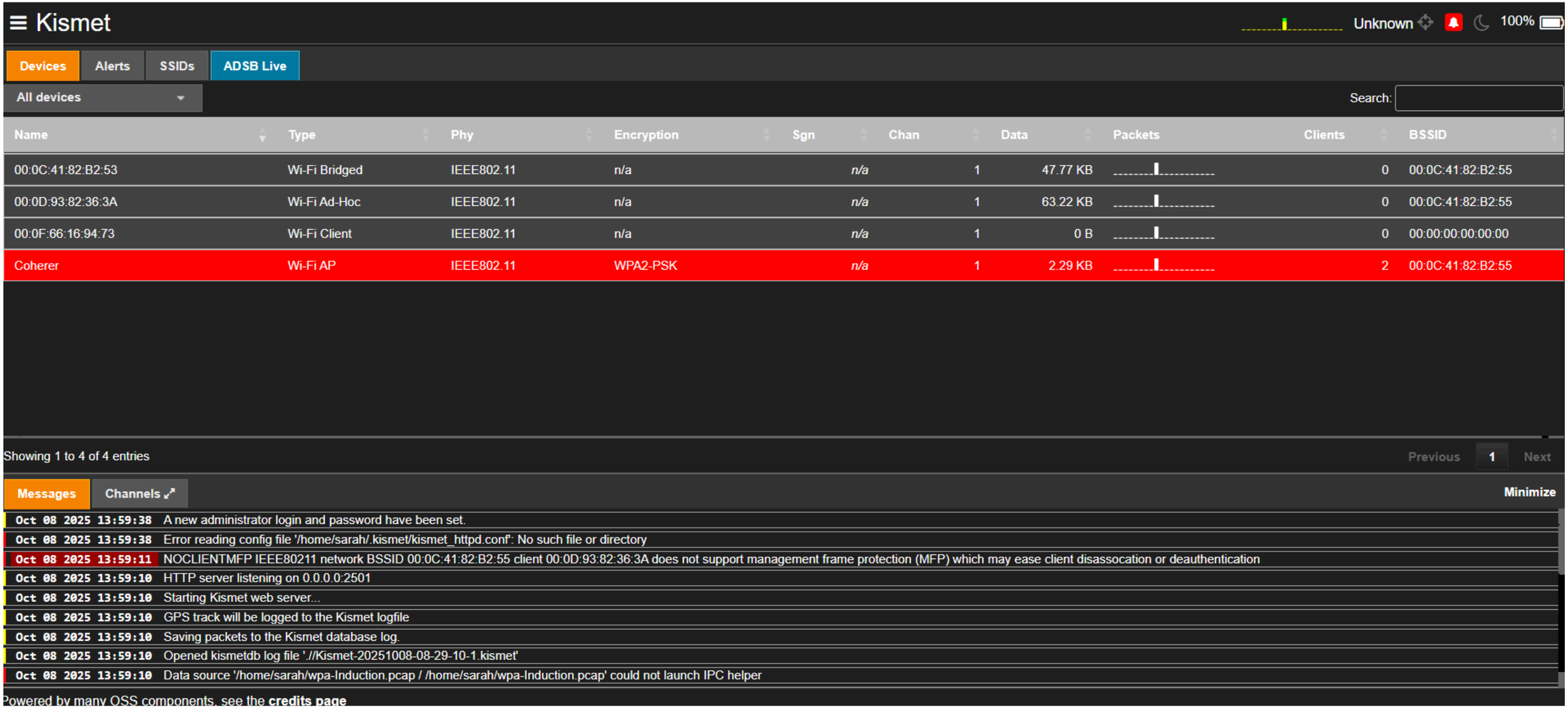
Running the wifi deauth.py file :

(Implements a Wi-Fi deauthentication attack tool that sends spoofed deauth packets to disconnect clients from a target Wi-Fi access point)

```
git clone https://github.com/flashnuke/wifi-deauth.git
```

```
python3 wifi_deauth.py
```

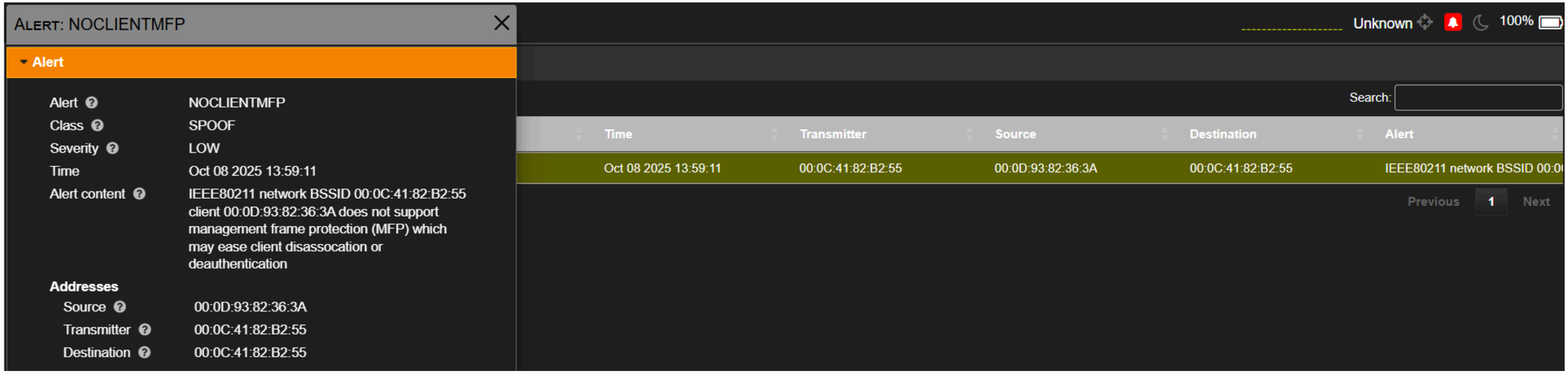
Kismet - Devices connected to that wi-fi



Interpreting Dashboard Results and Logs

- View alert details in the Alerts/Events tab for summary and breakdown of suspicious actions.
- Logged events include:
  - Type of attack detected (e.g., deauth flood)
  - Affected network and client MAC addresses
  - Number and timing of detected packets
  - Severity and recommended action
- Use dashboard filters and export options for deeper analysis or reporting.

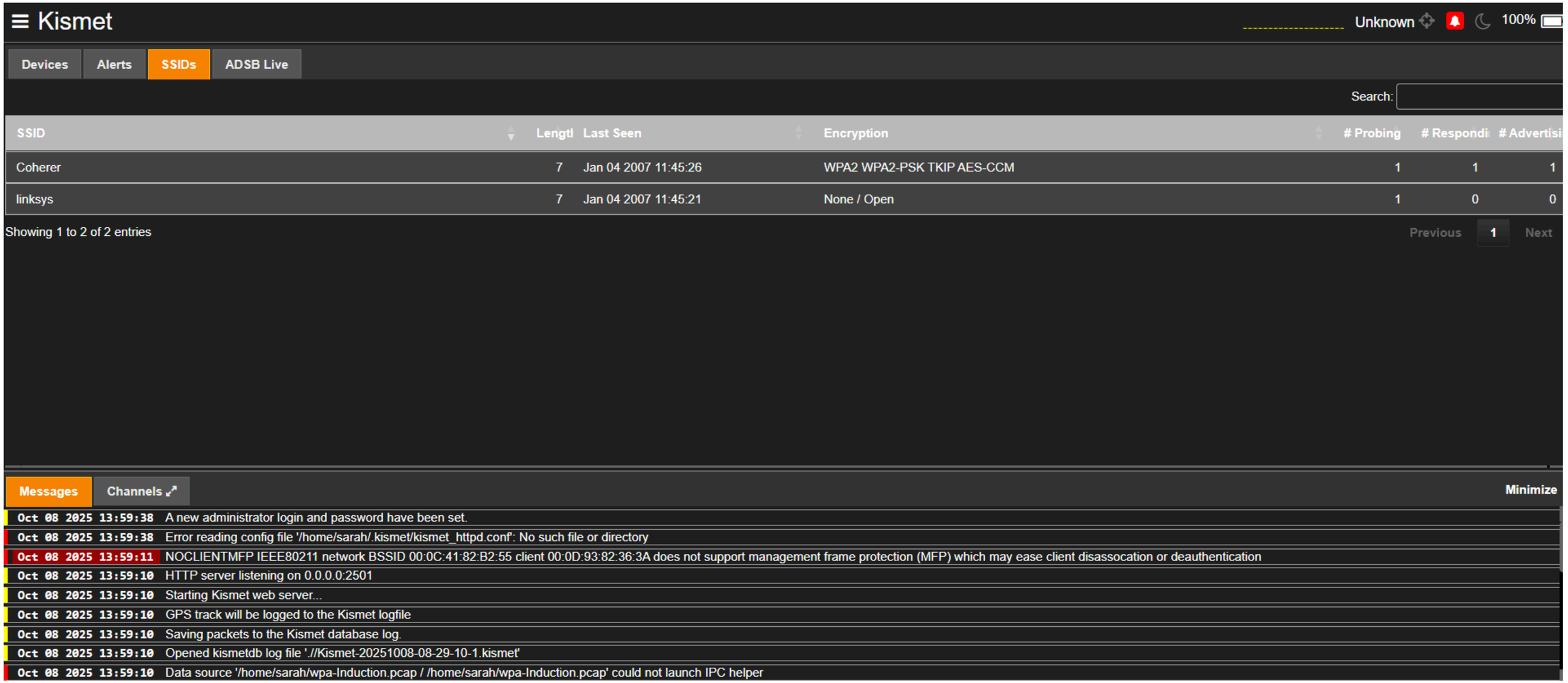
Analyze Attacks:



Understanding Kismet's Alerts

- NOCLIENTMFP: Indicates a client or network does not support management frame protection (MFP); these devices are more vulnerable to deauth/disassoc spoof attacks.
- Deauth Flood: Alert for multiple deauthentication packets detected, signaling a probable attack attempt.
- Other Alerts: Disassociation attacks & rogue AP detection. All provide details like MAC addresses, event times, and severity levels.

SSID (Probing, Responding, Advertising)



How to Prevent Deauth Attacks

- **Enable Management Frame Protection (MFP):** Use 802.11w standard that encrypts management frames to prevent spoofed deauth/disassoc attacks.
- **Use WPA3 Security Protocol:** The latest Wi-Fi encryption improves protection against various attacks, including deauthentication.
- **Strong Wi-Fi Passphrases:** Use complex, unique passwords to reduce unauthorized access and rogue devices.
- **MAC Address Filtering:** Allow only authorized devices to connect, limiting attacker access.
- **Disable Unused Features:** Turn off unused wireless features like WPS and SSID broadcast to minimize attack surface.
- **Keep Router Firmware Updated:** Patch known vulnerabilities regularly.

THANK YOU